計算科学演習

スーパーコンピュータ＆並列計算概論

学術情報メディアセンター
情報学研究科・システム科学専攻
中島　浩

目次

- 科目概要
  - 目標・スケジュール・スタッフ・講義資料・課題
- スーパーコンピュータ概論
  - 一般のスーパーコンピュータ
  - 京大のスーパーコンピュータ
  - スーパーコンピュータの構造
- 並列計算概論
  - 並列計算の類型・条件
  - Scaling & Scalability、問題分割、落し穴
  - プロセス並列 & スレッド並列、バリア同期
  - バッチジョブ

科目概要

目標・スケジュール・スタッフ・講義資料

- 目標
  - 拡散方程式の初期値求解問題を題材に
  - MPI と OpenMP を用いた並列プログラムを作成して
  - 並列プログラミングの基礎と(やや高度な)応用を学ぶ
- スケジュール
  - 第1日：スパコン＆並列計算概論(中島)+リテラシ演習(木村・高橋)
  - 第2日：拡散方程式＆陽解法(岩下)+逐次P作成(木村・高橋)
  - 第3日：MPI 基礎(中島)+1D分割P作成(木村・高橋)
  - 第4日：MPI 発展(中島)+2D分割P作成(木村・高橋)
  - 第5日：OpenMP基礎(岩下)+WS型P作成(木村・高橋)
  - 第6日：OpenMP発展(岩下)+領域分割型P作成(木村・高橋)
  - 第7日：レポート課題の仕上げ(中島・岩下・木村・高橋)
  - (第8日，第9日，…：9/30までは頑張れる)
- 講義資料http://ais.sys.i.kyoto-u.ac.jp/~iwashita/kougi-CSE.htm

科目概要

課題

- 課題＝拡散方程式の初期値求解 by 陽解法
  - C or Fortran + MPI / OpenMP
  - 課題1：逐次プログラム
  - 課題2(1)：MPI + 1次元分割
  - 課題2(2)：MPI + 2次元分割
  - 課題3(1)：OpenMP + Work Sharing
  - 課題3(2)：OpenMP + 領域分割
- 提出物・提出先・期限
  - 作成したプログラム5種(以上)のソースファイル
  - 課題内容・手法説明・プログラム概要・結果考察のレポート (MS Word or PDF)
  - h.nakashima@media.kyoto-u.ac.jp
    iwashita@media.kyoto-u.ac.jp
  - 9月30日(水) 17:00 必着

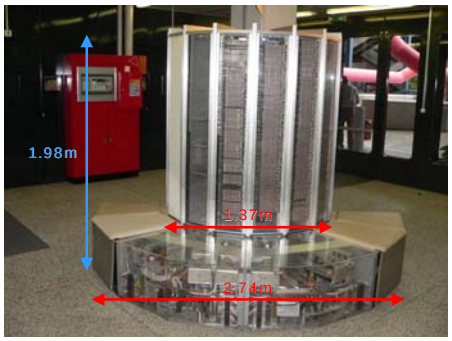
スーパーコンピュータ概論

一般のスパコン：ベクトルマシン (1/2)

- 1976年：最初のスパコンCray-1登場
  - 動作周波数=80MHz (< 携帯電話)
  - 演算性能=160MFlops (< 携帯電話)
    - Flops: Floating-point Operation Per Second
      = 10進16桁精度の数値(10^{-308}~10^{308})の加減乗算回数／秒
      →160MFlops = 毎秒1億6千万回の加減乗算
  - 消費電力=115kW
  - 大量の数値データ(ベクトル)に対する同種演算が得意
- その後
  - 1980年代：スパコン≒ベクトル(並列)マシンの時代
  - 1990年代：スカラ並列マシン(後述)との激闘
  - 2002年：地球シミュレータが7年振りにベクトルで最速に
  - 現在：地球#2がTOP500(後述)に唯一(22位)

スーパーコンピュータ概論

一般のスパコン：ベクトルマシン (2/2)



source: http://en.wikipedia.org/wiki/Image:Cray-1-p1010221.jpg

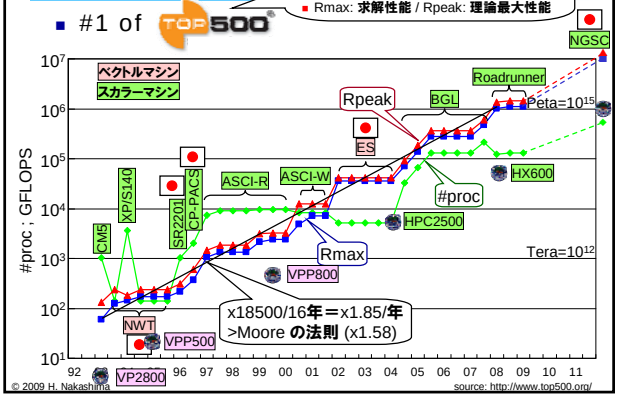
スーパーコンピュータ概論
一般のスパコン：スカラ並列マシン

- 1980年代に出現
  - Sequent Balance : 20 x NS32016 ('84)
  - Intel iPSC/1: 128 x i80286 ('85)
- 多数のパソコン(のようなもの)の集合体
  - 個々の部品(CPU, メモリなど)≒パソコン(＆ゲーム機)
    - 実際にTOP500(後述)では …
      x86 = 436(87.2%) v.s. others = 64(12.8%)
  - ただしメチャクチャに数が多い
    - パソコン = 1~4 CPU
    - 京大スパコン = 6,656 CPU
    - 世界最高速スパコン = 12,960 CPU + 12,960 PS3(+α)
    - 世界最大規模スパコン = 294,912 CPU
  - 同じような計算の集合体としての巨大計算が得意

スーパーコンピュータ概論
一般のスパコン：T

- 巨大で(>100万元)密な連立一次方程式の求解性能に基づく世界中のスパコン順位表
- 1993.6から毎年2回発表 (6月&11月)
- Rmax: 求解性能 / Rpeak: 理論最大性能

- #1 of TOP500



source: http://www.top500.org/

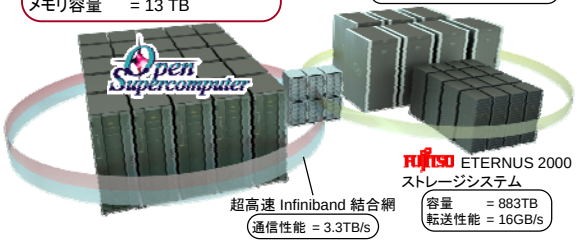
スーパーコンピュータ概論
京大のスパコン (1/2)

FUJITSU HX600 クラスタ

ノード数 = 416
コア数 = 16 x 416 = 6656
ピーク性能 = 61.2 TFlops
Linpack 性能 = 50.5 TFlops(#78)
メモリ容量 = 13 TB

FUJITSU SPARC Enterprise M9000 fat node サブシステム

ノード数 = 7
コア数 = 128 x 7 = 896
ピーク性能 = 8.96 TFlops
メモリ容量 = 1TB x 7 = 7 TB



超高速 Infiniband 結合網
通信性能 = 3.3TB/s

FUJITSU ETERNUS 2000 ストレージシステム
容量 = 883TB
転送性能 = 16GB/s

スーパーコンピュータ概論
京大のスパコン (2/2)

- 日本で第 9 位／世界で第 78 位の性能
- パソコンなどと比べると …

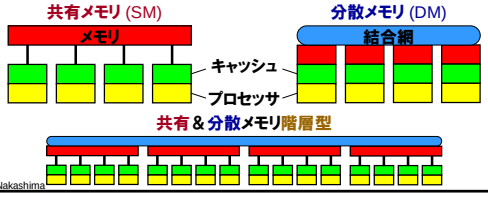
毎秒61兆回の加減乗算

	京大スパコン	パソコン	倍率
演算性能	61.2 TFlops	10 GFlops	x 6120
メモリ容量	13 TByte	2 GByte	x 6656
通信性能	3.3 TByte／秒	12.5 MByte／秒 (B-flets)	x 264000
ディスク容量	883 TByte	250 GByte	x 3532

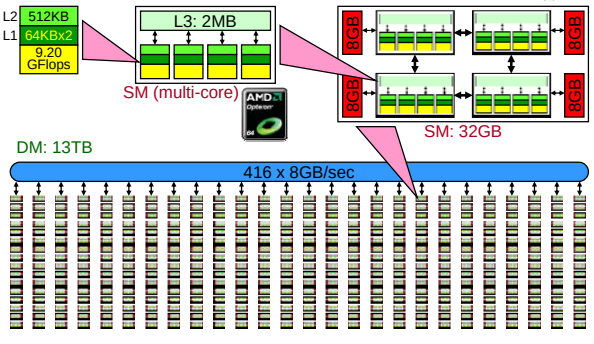
Tera = 10^{12} = 1兆
Giga = 10^{9} = 10億
Mega = 10^{6} = 100万

スーパーコンピュータ概論：スパコンの構造
共有メモリと分散メモリ

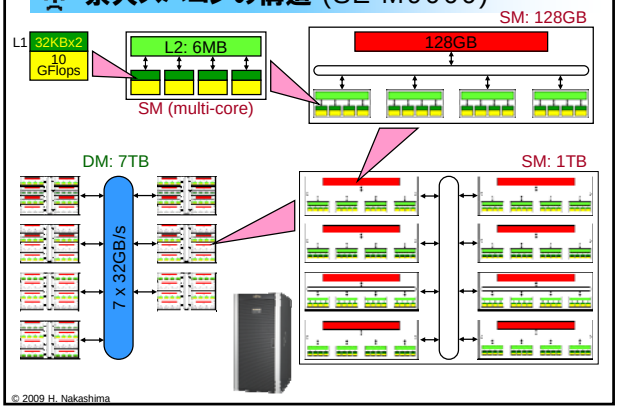
- 共有メモリ型
  - (論理的に)1つのメモリをプロセッサが共有➔変数共有可能
    ➔あるプロセッサが代入した値を別のプロセッサが参照可能
  - 一般に小規模(プロセッサ数＝10^0~10^2のオーダー)
- 分散メモリ型
  - 別々のコンピュータをネットワークで繋いだもの
    ➔プロセッサ間のデータのやり取りには陽に「通信」が必要
  - 大規模な構成が比較的容易(~10^5のオーダー)
- 共有＆分散メモリ階層型：最近の主流



スーパーコンピュータ概論：スパコンの構造
京大スパコンの構造 (HX600)



スーパーコンピュータ概論：スパコンの構造
京大スパコンの構造 (SE M9000)



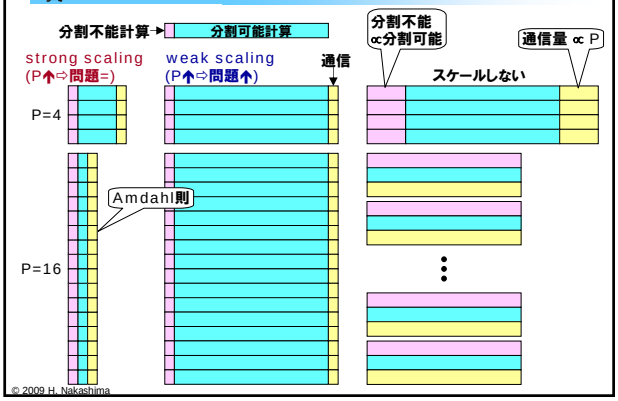
並列計算概論
並列計算の類型

- スパコンを使って計算をする理由＝高速計算
  - 高速なプロセッサを使う＝(特に今後は)困難
  - 多数のプロセッサを使う＝(昔も今後も)可能
- 問題 X の逐次実行時間 T(X,1)
  P 個のプロセッサでの並列実行時間 T(X,P)
  - T(X,P)≒T(X,1)／P　➔strong scaling
  - X^P＝X の P 倍の規模(メモリ量、計算量など)の問題
    $T(X^P,P)≒T(X^P,1)／P$　➔weak scaling
  - $X_1, \ldots, X_P$＝X の P 個のインスタンス
    $T(\{X_1,\ldots,X_P\},P)≒T(X,1)$ ➔capacity computing

並列計算概論
並列計算の条件

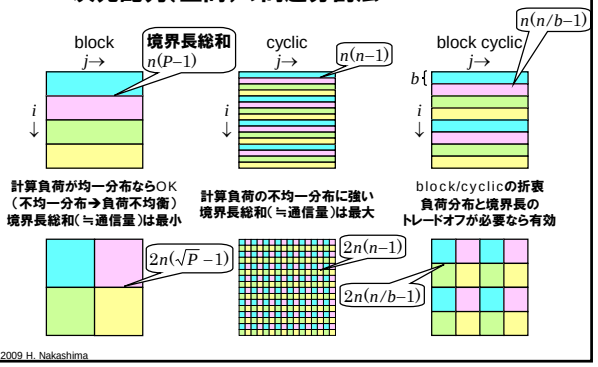
- P 倍程度の性能を得るための必要条件
  - 問題を計算量が 1/P 程度の部分問題に分割可能
  - 部分問題の必要メモリ≦問題の必要メモリの k/P (＋α)
    (特に weak scaling で重要)
  - 分割不能計算量 X^{seq} ≪ 分割可能計算量 X^{para}
    (strong scaling の一般的な限界)
  - 部分問題について通信時間／計算時間＜O(1)
- P 並列計算時間の粗い見積
  - $T(X,P)$ ≒ $T(X^{seq},1)$ ＋ $T(X^{para},1)/P$ ＋ 通信時間
  - 通信時間 ≒ 通信データ量／B ＋ 通信回数×L
    - B：バンド幅 ≒ 1~4GB/s
    - L：遅延＋オーバヘッド ≒ 5~50μs

並列計算概論
Scaling & Scalability



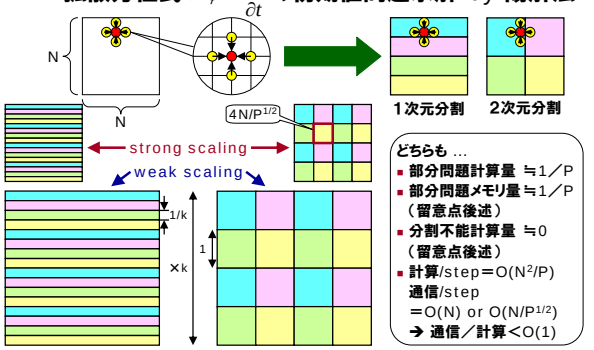
並列計算概論
問題分割 (1/2)

- 2次元配列(空間)の問題分割法



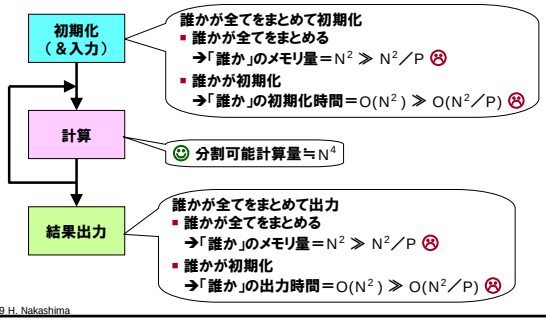
並列計算概論
問題分割 (2/2)：拡散方程式では…

- 拡散方程式 $\nabla^2\varphi=\frac{\partial\varphi}{\partial t}$ の初期値問題求解 by 陽解法



並列計算概論
拡散方程式プログラムの落し穴

- 部分問題メモリ量 ≒1／P ＆
  分割不能計算量 ≒0 ？



並列計算概論
プロセス並列 ＆ スレッド並列 (1/2)

- プロセス並列 (aka Message Passing) v.s.
  スレッド並列 (aka Shared Memory)
  - プログラミング・パラダイムとしての分類（≠H/W の分類）
  - ライブラリ／言語のコンセプト

	プロセス並列(PP)	スレッド並列(TP)
並列実行単位	プロセス	スレッド
アドレス空間	（プロセスに）固有	スレッド間で共有
通信（＆同期）	通信用プリミティブ (e.g. send, recv)	共有変数アクセス ＋同期プリミティブ
ライブラリ	(e.g.) MPI	(e.g.) OpenMP
H/W=DM	◎	△ (S/W DSM, 言語)
H/W=SM	○	◎
折衷型	ハイブリッド並列	

計算の歩調合せ

実質的には×

並列計算概論
プロセス並列 ＆ スレッド並列 (2/2)

- PP v.s. TP のアナロジー
  - PP ≒ メールの添付ファイル
    - データ転送の主導者 = 生産者
    - 受信データ = 必要なデータ

「ファイルを更新したので添付します」

  - TP ≒ web page（にリンクされたファイル）
    - データ転送の主導者 = 消費者
    - 獲得データ = 必要なデータ … とは限らない
    - ➔生産者➔消費者の同期が必要
    - ➔消費者➔生産者の同期も必要

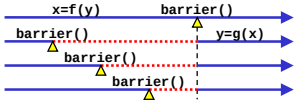
web を見ても更新の有無は不明

「download したので更新してもいいよ」

「ファイルを更新したので download してね」or web に更新日付を記載

並列計算概論
バリア同期

- スレッド並列プログラミングでの同期操作の定番
  - 全員がバリア(e.g. ループ終了)に到達するまで待つ
  - バリア前に更新した変数 ➔バリア後に安全に参照
    バリア前に参照した変数 ➔バリア後に安全に更新



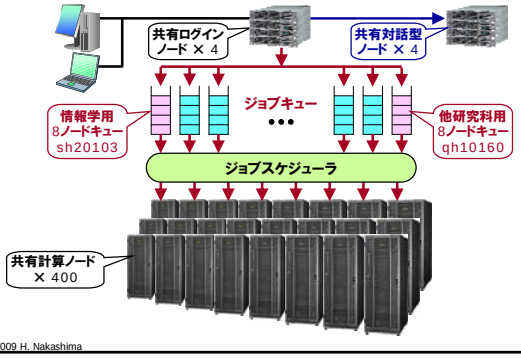
  - OpenMP では並列ループの終了時に自動バリア
  - プロセス並列でも使用することがある
    - （論理的ではなく）性能的な歩調合せ
    - ファイル I/O などの通信以外の協調動作のための同期

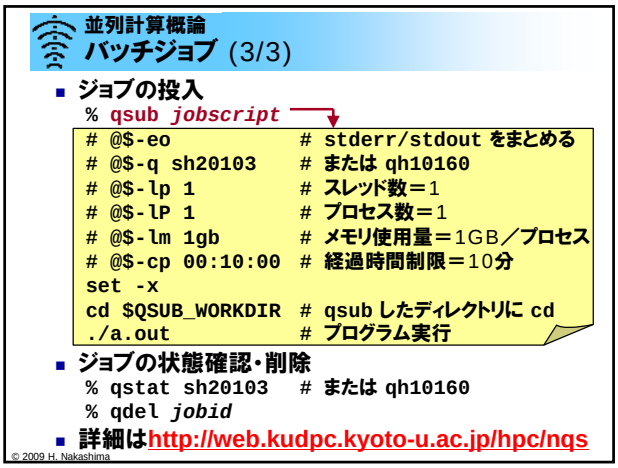
並列計算概論
バッチジョブ (1/3)

- 普通の実行＝interactive（対話的）
  - コマンド入力／クリックでプログラムが直ちに起動
    ←（普通は）計算資源を占有＆余裕あり
  - 実行中にプログラムと対話（キー入力・クリック）
    - 例外：virus scan, desktop検索, update, …
- スパコンでの実行＝batch
  - ジョブ(Prog+Data)実行を依頼 ≒願書郵送
  - 資源があれば／空けば実行開始 ≒担当者開封
  - (普通は長時間)非対話的に実行 ≒受験票到着
    ←計算資源は共有＆余裕僅少
    - 例外：edit, コンパイル，小規模テスト，ジョブ投入，…

並列計算概論
バッチジョブ (2/3)

- 京大スパコンのバッチジョブスケジューラ




並列計算概論
バッチジョブ (3/3)
ジョブの投入
% qsub jobscript
# @$-eo # stderr/stdout をまとめる
# @$-q sh20103 # または qh10160
# @$-lp 1 # スレッド数＝1
# @$-lP 1 # プロセス数＝1
# @$-lm 1gb # メモリ使用量＝1GB／プロセス
# @$-cp 00:10:00 # 経過時間制限＝10分
set -x
cd $QSUB_WORKDIR # qsub したディレクトリに cd
./a.out # プログラム実行
ジョブの状態確認・削除
% qstat sh20103 # または qh10160
% qdel jobid
詳細はhttp://web.kudpc.kyoto-u.ac.jp/hpc/nqs
© 2009 H. Nakashima